Algorithms for Wavefront Reconstruction

Stefan Hagdahl

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Algorithms for Wavefront Reconstruction

Stefan Hagdahl

N° 5696

September 2005

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--5696--FR+ENG

Algorithms for Wavefront Reconstruction

Stefan Hagdahl*

Thème NUM — Systèmes numériques
Projets OTTO

Rapport de recherche n° 5696 — September 2005 — 28 pages

Abstract: Knowing the solution to Helmholtz equation in a narrow band, with a width of order wave length around a scatterer, we investigate the possibility to construct the wave fronts predicted by the asymptotic methods Geometrical Optics (GO) and Geometrical Theory of Diffraction (GTD). More specifically we show how to build a polynomial approximation of the asymptotic solution on a circle around the scatterer and hence, in a second step, be able to propagate this solution by wavefront construction methods or standard ray-tracing.

Key-words: Geometrical Optics, Wavefront construction

* stefanh@nada.kth.se

Algorithms for Wavefront Reconstruction

Résumé : Etant donné la solution de l'équation de Helmholtz dans une bande de largeur de l'ordre d'une longeur d'onde autour d'un obstacle diffractant, nous examinons la possibilité de construire les fronts d'ondes prédits par l'optique géométrique (GO) et la théorie géométrique de la diffraction (GTD). Plus précisement nous montrons comment construire une approximation polynomiale de la solution asymptotique sur un cercle entourant l'obstacle et donc, dans un second temps nous sommes capable de propager cette solution en utilisant une méthode de fronts d'ondes ou le lancer de rayon standard.

Mots-clés : Optique Géométrique, Wavefront construction

Contents

1 Introduction

We start with the derivation of the Geometrical Optics (GO) model. Let $u(x)$ be the solution to Helmholtz equation

$$\triangle u + \left(\frac{2\pi\eta}{\lambda_0}\right)^2 u = 0,$$

supplemented by suitable boundary and radiation conditions. The coefficient $\eta = \eta(x)$ is the relative index of refraction and $\lambda_0 = 2\pi/k$ is the wavelength in vacuum for a wave with wave number k. Below we will only consider scattering problems in vacuum ($\eta = 1$) and we will therefore suppress the index 0. One of the most basic expansions in Geometrical Optics relies on the idea that the complex valued solution $u_\lambda(x)$ can be approximated, asymptotically in λ, by the "ansatz"

$$u_\lambda(x) = e^{i2\pi\phi(x)/\lambda} \sum_{p=0}^{n} A_p \lambda^p, \tag{1}$$

where the amplitudes $A_p(x)$ and the phase $\phi(x)$ are frequency independent real valued solutions of the Eikonal/transport GO system of equations,

$$|\nabla\phi| = 1 \tag{2}$$

and

$$2\nabla\phi \cdot \nabla A_1 + A_1 \Delta\phi = 0 \qquad \mathcal{O}(\lambda)\text{-equation} \tag{3}$$

$$2\nabla\phi \cdot \nabla A_p + A_p \Delta\phi = -\Delta A_{p-1} \quad \mathcal{O}(\lambda^p)\text{-equation.} \tag{4}$$

Expansion (1) can be generalized to take into account several phases or branches. This report will present some analysis and suggest some algorithms to find polynomial approximations of the phases and amplitudes when the Helmholtz solution is known in a narrow band, with a width of $\mathcal{O}(\lambda)$, around a scatterer.

One of the major challenges in the algorithms is to be able to detect the curves where the number of phases, or branches, changes. The total asymptotic solution is generally discontinuous along these curves. By mathematical analysis one can show that these curves are the so called *caustic curves*. The Numerical MicroLocal Analysis (NMLA) presented in [1] are, since the asymptotic solution is discontinuous, not applicable in the vicinity of the curves. We will therefore limit our analysis of the wavefront reconstruction algorithms along curves that are approximately non-parallel to the caustic curves.

1.1 Regularity and Polynomial Expansions

Since (2) is a non linear PDE there exist no classical global unique solution in general. However under specific conditions there are quite strong bounds on ϕ, its derivatives and the A_ps. Under some conditions it is also possible to assume uniqueness. However, below we will only assume that:

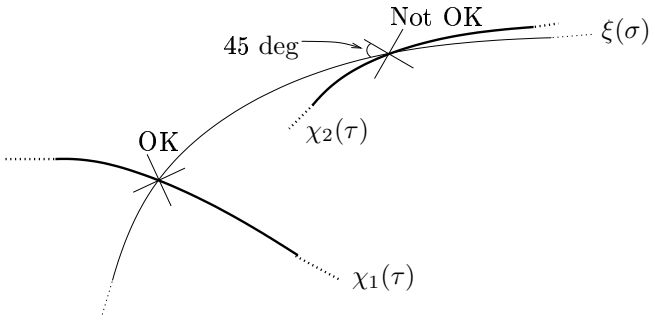

Figure 1: Illustration of caustics that assumption (5) allow and do not allow.

1. there is a finite number of solutions in all points
2. ϕ, $\nabla\phi$ and A_p for *each* solution is continuous.
3. the caustic curves $\chi_k(\tau)$ which split domains with different number of solutions, let us say $\phi^L_{1,k} \dots \phi^L_{j,k} \dots \phi^L_{m_L,k}$ solutions on one side and $\phi^R_{1,k} \dots \phi^R_{m_R,k}$ solutions on the other side, fulfill

$$\arccos\left(\frac{\nabla\phi_{j,k} \cdot \frac{\partial\chi_k}{\partial\tau}}{|\nabla\phi_{j,k}||\frac{\partial\chi_k}{\partial\tau}|}\right) < \alpha \leq \frac{\pi}{4} \quad \forall\ (j,k)\text{-pairs and } |x| \geq R - \lambda.$$

Our analysis and algorithms relies on the following extension to the ideas presented in [1]:

By assumption 2 to 3 we can assume that the locally plane waves detected in a point x_q, by the transforms presented in [1], may be parameterized along a curve $\xi_q(\sigma)$ which fulfill

$$\max_\tau \arccos\left(\frac{\frac{\partial\xi_q}{\partial\sigma} \cdot \frac{\partial\chi_k}{\partial\tau}}{|\frac{\partial\xi_q}{\partial\sigma}||\frac{\partial\chi_k}{\partial\tau}|}\right) \geq \frac{\pi}{4} \text{ when } |x| \geq R, \tag{5}$$

for all τ and k which fulfill $\min_\sigma |\xi_q(\sigma) - \chi_k(\tau)| < \mathcal{O}(\lambda)$. In Figure 1 we illustrate this constraint.

The constraints above will, as we will discuss later on, guarantee that we can use the transforms in [1] a number of times to construct all wavefronts along a curve $\xi(\sigma)$. The non-planar nature of the GO-waves may then be captured by combining transforms in some points along a segment of $\xi(\sigma)$ and build polynomial approximations ϕ_j and A_{pj}, where j is a branch index, on that segment.

Hence we assume that the solution to Helmholtz equation asymptotically can be written as

$$u_\lambda(\xi(\sigma)) = \sum_{j=1}^{m(\sigma)} e^{i\frac{2\pi}{\lambda}\phi_j(\sigma)} \left(\sum_{p=0}^{n} A_{pj}(\sigma)\lambda^{e[p,j]}\right), \tag{6}$$

where each ϕ_j solves the Eikonal equation and each A_{pj} solves the pth transport equation. With the multi index functions $e\,[p,j]$ in the exponent of λ we allow for solutions that are not necessarily algebraically dependent on the wave length. From now on, let us only consider the terms with $i=0$. We therefore suppress the p-index and and introduce the following notation

$$\tilde{u}_\lambda(\sigma) = \sum_{j=1}^{m(\sigma)} e^{i\frac{2\pi}{\lambda}\phi_j(\sigma)} A_{0j}(\sigma)\lambda^{e[0,j]} = \sum_{j=1}^{m(\sigma)} \bar{u}_j(\sigma).$$

According to above we assume that $\phi_j(\sigma)$ and $A_j(\sigma)$ are continuous along a curve $\xi(\sigma) \leftrightarrow \bigcup_q \xi_q(\tilde{\sigma})$ but the maximal summation index m depends on σ, i.e. $m(\sigma)$.

For later reference we note that if we know[1] $\phi_{,1}$ we can, by using (2), compute $\phi_{,2}$ up to a sign, i.e.

$$\phi_{,2} = \pm\sqrt{1-\phi_{,1}^2}. \tag{7}$$

Assuming that we know one of the derivatives in

$$[\phi_{,1,1}, \phi_{,2,2}, \phi_{,1,2}], \tag{8}$$

then we can, by differentiating the Eikonal equation with respect to x_1 and x_2, and solving the corresponding linear system, compute the rest of the derivatives in (8).

1.2 Eikonal and Transport Equation in Curvelinear Coordinates

Let us consider one phase or branch in (6).

The analysis in Section 1.1 can also be applied to curve linear coordinates. If we know $\phi_{,\sigma}(\sigma)$ and $\phi_{,\sigma,\sigma}(\sigma)$ along a given curve $\xi(\sigma)$ we can, by using the chain rule, compute any Cartesian derivative up to order 2. Below we will set $\xi(\sigma)$ to be the circle in $\mathbb{R}^2$. Referring to Figure 2 we use the following notation to parameterize the circle

$$\gamma := \{\xi(\theta) \in \mathbb{R}^2 : \theta \in [0, 2\pi], |\xi(\theta)| = R\}. \tag{9}$$

Let $\phi(x_1(r,\theta), x_2(r,\theta))$ then by the chain rule we compute

$$\phi_{,\theta} = \phi_{,1}\frac{\partial x_1}{\partial\theta} + \phi_{,2}\frac{\partial x_2}{\partial\theta}. \tag{10}$$

We note the following for later reference. If ϕ and $\nabla\phi$ is known in some points along a circular curve segment $\xi_q(\theta)$, starting in x_q, then an approximate $\phi(\theta)$ can be constructed by using, for example, the power basis in polynomial curve fitting. Similar holds for the A_js. Hence, given a polynomial approximation of $\phi(\theta)$ in curve segment $\xi_q(\theta)$, with

$$\phi_q(\theta) = \sum_{\ell=0}^{\kappa_\phi} c_{\ell,q}^{(\phi)}\theta^\ell, \tag{11}$$

[1] We use Einstein notation for derivatives by putting a comma in front of the index. Comma in front of an integer means derivative with respect to a Cartesian coordinate.

derivatives such as $\phi_{,\theta}$ and $\phi_{,\theta,\theta}$ is easily accessible. In (11) and in similar expressions below we will suppress the fixed radius R. For convenience we will call $\beta = \text{angle}(\nabla_x \phi)$ where angle is a function returning the angle $\beta = [0, 2\pi)$ between the gradient and $\hat{x}_1$.

C.f. Figure 2 and definition (9). If we let γ encircle the scatterer S positioned close to the origin, R can be chosen sufficiently large so that the sign ambiguity in (7) can be solved by the natural assumption that the scalar product between the gradient of the scattered waves and the normal $\hat{n}_\gamma$ are non-negative, i.e.

$$\cos\beta(\theta, r)\cos\theta + \sin\beta(\theta, r)\sin\theta \geq 0. \tag{12}$$

Note that the assumption (12) is consistent with Sommerfeldt's radiation condition as $R \to \infty$.

We can also, by coordinate transformation, achieve an Eikonal equation in polar coordinates,

$$\left| (\phi_{,r})^2 + \left(\frac{\phi_{,\theta}}{r} \right)^2 \right| = 1. \tag{13}$$

Hence, by using the chain rule we can with (10), (12) and (13), i.e.

$$\phi_{,\theta} = \sqrt{1 - \left(\frac{\phi_{,\theta}}{r} \right)^2}, \tag{14}$$

compute β along γ according to

$$\phi_{,k} = \sqrt{1 - \left(\frac{\phi_{,\theta}}{r} \right)^2} r_{,k} + \phi_{,\theta}\theta_{,k} \tag{15}$$

and

$$\beta(\theta) = \text{angle}(\nabla_x \phi). \tag{16}$$

Also by differentiating (13) with respect to r and θ and once more using (12), we can solve the following linear system for $\phi_{,r,r}$ and $\phi_{,r,\theta}$

$$\begin{pmatrix} r^3\phi_{,r} & r\phi_{,\theta} \\ 0 & r^2\phi_{,r} \end{pmatrix} \begin{pmatrix} \phi_{,r,r} \\ \phi_{,r,\theta} \end{pmatrix} = \begin{pmatrix} \phi_{,\theta,\theta}^2 \\ -\phi_{,\theta}\phi_{,\theta,\theta} \end{pmatrix} \tag{17}$$

If we numerically want to compute $\beta(\theta)$ for a specific β then we can of course use (11) and (16). However, if we search β for many values then it may be practical build a $\beta(\theta)$-polynom,

$$\beta_q(\theta) = \sum_{\ell=0}^{\kappa_\beta} c_{\ell,q}^{(\beta)} \theta^\ell, \tag{18}$$

Similar holds for the inverse of the Laplacian of ϕ, which plays an important role when solving the Transport equations by for example ray tracing. That is, it is practical to build a polynom of

$$\rho = \frac{1}{\triangle\phi}, \tag{19}$$

as

$$\rho_q(\theta) = \sum_{\ell=0}^{\kappa_\rho} c_{\ell,q}^{(\rho)} \theta^\ell. \tag{20}$$

The coefficients can then be set by combining (17) and the inverse of the Laplacian in polar coordinates

$$\rho(\theta) = \frac{R^2}{R^2\phi_{,r,r} + \phi_{,\theta,\theta} + R\phi_{,r}}, \tag{21}$$

in some points $\theta_i, i = 1 \dots \kappa_\rho$.

Finally we state for further reference the polynomial expansion of the amplitude,

$$A(\theta) = \sum_{\ell=0}^{\kappa_A} c_\ell^{(A)} \theta^\ell. \tag{22}$$

2 Interpolation

In Section 1.2 we indicated how $A(\theta)$, $\phi(\theta)$, $\beta(\theta)$ and $\rho(\theta)$ can be accessed by evaluating a polynom. Naturally this introduce interpolation errors. These errors can be bounded by error estimates from the literature. However to proceed we need to discuss the errors inherited by the transforms presented in [1].

From the analysis and the numerical experiments in [1] it is clear that the errors in the numerically computed GO/GTD-coefficients $\bar{u}_j = e^{ik\phi_j} A_j$ from the Numerical MicroLocal Analysis are, after optimization, of order

$$\max_j \mathcal{O}(\lambda^{e[1,j]}) \equiv \mathcal{O}(\lambda^{\mathrm{me}}). \tag{23}$$

Then me depends on the scatterer, the excitation and the position of the NMLA point. We can for example have the following *cases*:

1. If no diffraction is present then the error is of $\mathcal{O}(\lambda)$
2. With only edge diffraction present the error is of $\mathcal{O}(\lambda^{3/2})$ (c.f. [2])
3. With both GO and diffraction present but only[2] the GO-component taken into account the error is of $\mathcal{O}(\lambda^{1/2})$.

The errors of β, and hence $\nabla_x \phi$, can also be seen to be, after optimization and with no diffraction present, of $\mathcal{O}(\lambda)$ in [1].

By using NMLA in a point x_q we can numerically estimate the phase function by

$$\bar{\phi}_j(x_q) = \lambda \frac{\log \bar{u}_j(x_q) - \log |\bar{u}_j(x_q)|}{2\pi\sqrt{-1}} = \phi_j(x_q) + \omega_j + \mathcal{O}(\lambda^{\mathrm{me}+1}), \tag{24}$$

[2] This could happened because of numerical errors.

where ω_j is an unknown integer and we used standard Taylor expansions to estimate the error. That is, compared to $\bar{u}_j$ we have gained one order of λ in accuracy but we have no precision! However, the error in precision is not of importance since we are in the end only interested in the asymptotic solution u_λ of Helmholtz equation.

The error in the amplitude A can straight forwardly be found to be of $\mathcal{O}(\lambda)$.

Consider case 1. As previously mentioned we can compute $\bar{\phi}_j$ in a certain number of points in an interval or curve segment of size $\triangle\theta$ and create an interpolating polynomial on that segment. In our implementation presented below we will assume segments of size $\triangle\theta = \mathcal{O}(\sqrt{\lambda})$. Then we can fit a cubic polynomial $\mathcal{P}^{(3)}(\theta)$ in three points according to

$$\begin{pmatrix} \mathcal{P}^{(3)}_{(\phi)}(\theta) \\ \mathcal{P}^{(3)}_{(\phi),\theta}(\theta) \\ \mathcal{P}^{(3)}_{(\phi),\theta}(\theta+\triangle\theta/2) \\ \mathcal{P}^{(3)}_{(\phi),\theta}(\theta+\triangle\theta) \end{pmatrix} = \begin{pmatrix} \bar{\phi}(\theta) \\ \bar{\phi}_{,\theta}(\theta) \\ \bar{\phi}_{,\theta}(\theta+\triangle\theta/2) \\ \bar{\phi}_{,\theta}(\theta+\triangle\theta) \end{pmatrix}, \tag{25}$$

and we can compute

$$\left[c_0^\phi, c_1^\phi, c_2^\phi, c_3^\phi\right], \tag{26}$$

in (11) with $\kappa_\phi = 4$. The last three components on the right hand side of (25) is computed with (10) where we use the knowledge of $\beta = \nabla_x\phi$ in the NMLA points $[\theta, \theta+\triangle\theta/2, \theta+\triangle\theta]$. Since the interpolation error for a cubic polynomial is of $\mathcal{O}((\triangle\theta)^4)$ the error expressed in λ will be, by choosing $\triangle\theta = \mathcal{O}(\sqrt{\lambda})$, of the same size as for the NMLA-error in the GO-case (c.f. (23) and (24)), i.e.

$$\max\left(\text{Error NMLA}, \text{Error interpolation}\right) = \max\left(\mathcal{O}(\lambda), \frac{\mathcal{O}((\sqrt{\lambda})^4)}{\lambda}\right) = \mathcal{O}(\lambda). \tag{27}$$

In the algorithms below we have used collocation in the three NMLA-points to set the coefficients in $\mathcal{P}^{(2)}_\beta$, $\mathcal{P}^{(2)}_A$ and $\mathcal{P}^{(2)}_\rho$. The interpolation error then becomes

$$\max\left(\text{Error NMLA}, \text{Error interpolation}\right) = \max\left(\mathcal{O}(\lambda), \mathcal{O}\left(\left(\sqrt{\lambda}\right)^3\right)\right) = \mathcal{O}(\lambda), \tag{28}$$

when applying the β-, A- and ρ-polynom. Note that we could also compute β by differentiating $\mathcal{P}^{(4)}_\phi$, which would give us an error of the same size as in (28).

Finally we want to discuss the asymptotic and numerical errors we do in a point $x \in \mathbb{R}^2 \setminus \Sigma$. The idea with setting a set of asymptotic solutions $\bar{u}_j$s on the circle γ is that we then have initial conditions to the system of Transport equations given in (3) and (4). Hence to estimate the errors in x, we need to investigate the error propagation in the ODE-solver that we use. This is straight forward if the asymptotic solution is continuous along the circle, i.e. consist of a fixed number of phases, but we believe it is a non-trivial task to do the same analysis for discontinuous solutions. We conclude, for this reason, that it is very important in future work to localize, with high accuracy, the caustic points along the circle.

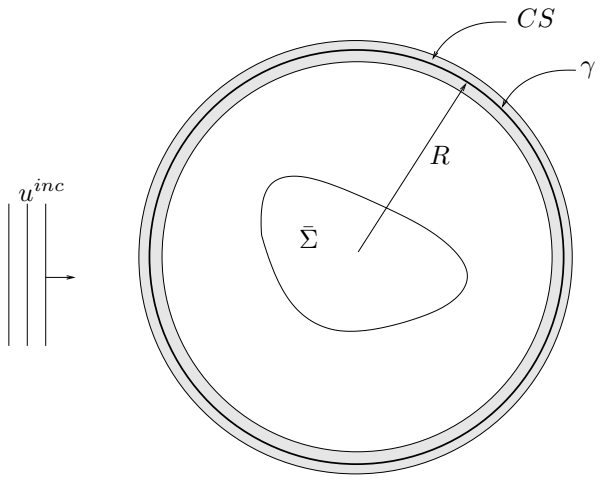

Figure 2: Problem setup.

3 Statement of Problem that the Algorithms Solve.

To present the algorithms that we used to reconstruct the asymptotic solution we here state the numerical problem more formally. We refer to Figure 2 and we make an $\mathcal{O}(\lambda^{\mathrm{me}})$ approximation of (6) as follows.

For a given scatterer Σ, an incoming wave u^{inc} and boundary conditions assume that we know Helmholtz solution u in the circular strip

$$CS := \{x \in \mathbb{R}^2 : R - \frac{\alpha\lambda}{2\pi} \leq |x| \leq R + \frac{\alpha\lambda}{2\pi}, 0 \leq \arcsin\left(\frac{x_1}{|x|}\right) \leq \pi\}. \tag{29}$$

Then, for a fixed radius R, α and variable $\theta = \mathrm{angle}(x)$, we search $m(\theta)$, $\phi_j(\theta)$ and $A_j(\theta)$ in

$$\tilde{u}_\lambda(\theta) = \sum_{j=1}^{m(\theta)} \bar{u}_j(\theta) = \sum_{j=1}^{m(\theta)} e^{ik\phi_j(\theta)} A_j \lambda^{e[0,j]}. \tag{30}$$

Hence, the $\bar{u}_j(\phi_j, A_j, \lambda)$ provide us with initial conditions to solve the ODEs in (3) and (4) in $\mathbb{R}^2 \setminus \Sigma$ by standard ray tracing or wavefront construction methods.

4 Algorithms

The NMLA-algorithm will be mentioned in Section 4.1 where we have investigated a new feature of the optimization algorithm already present in [1].

For smooth solutions, i.e. $m(\theta) \equiv m$, and case 1 in Section 2 we will later illustrate with some numerical experiments that our algorithms, presented in this section, fulfills

$$|\tilde{u}_\lambda - u|_{L^\infty} = \mathcal{O}(\lambda). \tag{31}$$

For non-smooth solution we will illustrate that the L^1-error is of $\mathcal{O}(\lambda)$ since the position of the discontinuities is localized within a distance of $\mathcal{O}(\lambda)$. The algorithms to achieve these asymptotic error bounds will be presented in Section 4.2 to 4.4.

β as well as ρ are important parameters when the asymptotic solution is to be propagated with an ODE-solver. The algorithms to find the $\mathcal{P}_\beta^{(2)}$ and $\mathcal{P}_\rho^{(2)}$ within an $\mathcal{O}(\lambda)$-error will therefore be mentioned in Section 4.5. In this section we will also mention how ray tracing easily can be used to propagate the solution in homogeneous materials.

Finally we note that the algorithms needs to be tuned with a set of parameters to produce satisfactory results.

4.1 Algorithm 1, Local Wavefront Reconstruction

We mainly refer to [1] for this part of the algorithm. We would like to emphasize though that there is an inherent weakness by the NMLA. Depending on which set of parameters that one uses, the algorithm can only distinguish between branches that are separated above a certain threshold in β-space. That is, with a parameter dependent ϵ, two branches with $|\beta_1 - \beta_2| < \epsilon$ will locally in a NMLA-point be considered as one branch. The wavefront reconstruction algorithm, which in some sense is global, is not able to correct this failure. However, as soon as $|\beta_1 - \beta_2|$ is large enough along γ the NMLA-algorithm will be called and hence the two branches may be distinguished.

We tried to improve the NMLA-algorithm as follows. We noted that, if there where only one branch present, i.e. $m \equiv 1$, it was possible to find the radius of curvature $\rho = (\triangle\phi)^{-1}$ of the wavefront, by using local non-linear optimization on ρ in the following analytic GO-solution[3],

$$\bar{u}_\lambda(x,\rho) = \bar{u}_\lambda(x_0)\sqrt{\frac{\rho}{\rho + (|Q(x)| - \rho)}}e^{ik(|Q(x)|-\rho)}, \tag{32}$$

where we assumed a circular wave starting at $x_0 - \rho\,[\cos\beta, \sin\beta]$ and we used the notation explained by Figure 3. Hence, we assume that the asymptotic field is given by a spherical wave and set $x = x(r,\delta) = x_r(\delta)$, then we can search

$$\min_\rho \int_0^{2\pi} \left| u(r,\delta) - \bar{u}_r^{\text{circ}}(r,\delta,\rho) \right| d\delta, \tag{33}$$

for example by applying a Gauss-Newton type minimization method with respect to ρ in

$$\min_\rho \sum_{\ell=0}^{M} \left| u(r,\ell\Delta\delta) - \bar{u}_\lambda(x_0) F_\ell(\rho;\delta,\beta,\lambda,r) \right|, \tag{34}$$

[3]This analytic GO-solution can be found by solving the Eikonal equation (2) and the first transport equation (3) for a point source. C.f. [3]

where

$$F_\ell(\rho;\delta,\beta,\lambda,r) = \left(\left|\tfrac{r}{\rho}\left[\cos\left(\ell\Delta\delta\right),\sin\left(\ell\Delta\delta\right)\right] - \left[\cos\left(\beta\right),\sin\left(\beta\right)\right]\right|\right)^{-1/2} \cdot \exp\left(\tfrac{i2\pi(|r[\cos(\ell\Delta\delta),\sin(\ell\Delta\delta)]-\rho[\cos(\beta),\sin(\beta)]|-\rho)}{\lambda}\right), \tag{35}$$

and β and $\bar{u}_\lambda(x_0)$ are given by the NMLA-algorithm. Above r is the λ-dependent circle-radius, M the number of points on the circle used in the NMLA-algorithm and F_ℓ is found by trigonometry. M and r are set by the following expressions

$$M = 2\cdot \text{round}\left(\alpha + C\sqrt[3]{\alpha}\right) + 1$$

and

$$r = \frac{\lambda\alpha}{2\pi}, \tag{36}$$

where we have set $C = 5$ and $\alpha = 2\pi$.

4.2 Algorithm 2, Polynomial Approximation of Wavefronts Along a Circle Segment.

Assume that we have found a circle segment S_q from a point

$$R\left[\cos(\theta_q),\sin(\theta_q)\right]$$

to

$$R\left[\cos(\theta_q+\Delta\theta_q),\sin(\theta_q+\Delta\theta_q)\right]$$

where $m(\theta_q) \equiv m$ for all θ_q in the interval. Then we apply the NMLA-algorithm in the three points

$$\begin{array}{l} R\left[\cos(\theta_q),\sin(\theta_q)\right] \\ R\left[\cos(\theta_q+\Delta\theta_q/2),\sin(\theta_q+\Delta\theta_q/2)\right] \\ R\left[\cos(\theta_q+\Delta\theta_q),\sin(\theta_q+\Delta\theta_q)\right], \end{array} \tag{37}$$

and compute the coefficients in $\mathcal{P}^{(4)}_{q,j,\phi}$ and $\mathcal{P}^{(3)}_{q,j,A}$ for all m branches. We use (10) and $\nabla_x\phi = [\cos(\beta),\sin(\beta)]$ to compute the right hand side in the last three equations in (25).

4.3 Algorithm 3, Adaptive Stepping Along a Circle.

Assume that we have applied Algorithm 2 to segment S_q and let c_1 and c_2 be two constants independent off λ. We find the size of following circle segment $S_{q+1} : \{\theta^{min}_{q+1}\ldots\theta^{max}_{q+1}\}$ by imposing the approximate error

$$e_{q+1}(\tilde{\theta}^{max}_{q+1}) = |\tilde{u}_\lambda - u|_{L^\infty,q+1} \approx c_1|\sum_{j=1}^{m}\bar{u}_j(\tilde{\theta}^{max}_{q+1}) - u(\tilde{\theta}^{max}_{q+1})| \approx c_2\lambda\min_j|\bar{u}_j(\theta^{min}_{q+1})|, \tag{38}$$

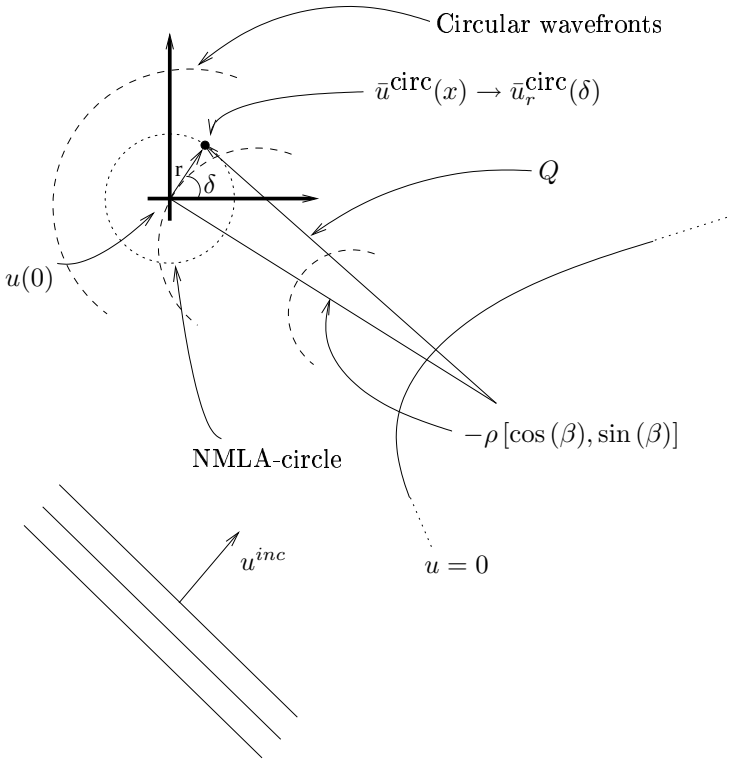

Figure 3: Local optimization of radius of curvature.

and θ_{q+1}^{max} is taken to be the point where approximately $\partial_\theta e_{q+1}$ attain its maximum in $\tilde{S}_{q+1} : \{\theta_{q+1}^{min} \ldots \tilde{\theta}_{q+1}^{max}\}$, i.e.

$$\max_{\ell} \frac{e_{q+1}(\theta_{\ell+1}) - e_{q+1}(\theta_\ell)}{\delta\theta}, \quad \text{with } \theta_\ell = \theta_{q+1}^{min} + \ell\delta\theta \leq \tilde{\theta}_{q+1}^{max}. \tag{39}$$

In (38) we use the polynoms in Algorithm 2 to approximate the asymptotic solution $\tilde{u}_\lambda$ at the end point of segment $\tilde{S}_{q+1}$ (i.e. we extrapolate the polynoms associated to segment S_q). The *adaptive* algorithm to find $\tilde{\theta}_{q+1}^{max}$ is more or less arbitrary but to minimize the computational cost[4] we propose that efficiency is prioritized.

The reason for using the bound in (38) is that we want to have a small segment if the ϕ_js and $|\bar{u}_j|$s change substantially and vice versa. The reason for using criteria (39) is that we want to detect the new branches that may show up discontinuously along γ. Of course if the

[4]If a boundary element solver is used to compute u then each evaluation will need $\mathcal{O}(\lambda^{-1})$ floating point operations.

new branches have a small amplitude then they will not be detected by the criteria. On the other hand if they are weak then they are probably not as important when the wavefronts are to be propagated.

When θ_{q+1}^{max} is set we want call NMLA in this point to be able to run Algorithm 2 for S_{q+1}. However, to decrease the risk that we call the NMLA in a point that are in the vicinity of a caustic curve we do the following.

Check for change in $m(\theta)$ by increasing θ_{q+1}^{max} according to

$$\theta_{q+1}^{max} \rightarrow \theta_{q+1}^{max} + \frac{2r}{R}, \tag{40}$$

where r is given by (36), and run the NMLA-algorithm. If there is no change in the number of branches compared to the NMLA in θ_{q+1}^{min} then we call Algorithm 4 and build new polynoms in S_{q+1} with Algorithm 2. If there is an increase in number of branches assume that they where introduced exactly at θ_{q+1}^{max} and build a first order approximation of the corresponding polynoms by using the NMLA-data in θ_{q+1}^{max}. Call Algorithm 4 and build new polynoms in S_{q+1} with Algorithm 2.

If there is a decrease in number of branches assume that they disappeared exactly at θ_{q+1}^{max} and set their corresponding amplitudes to 0 at θ_{q+1}^{max}. Call Algorithm 4 and build new polynoms in S_{q+1} with Algorithm 2.

Note that it is now natural to introduce the notation m_q since we have the same number of branches in the entire segment S_q. Also note, that by assumption (5) it is reasonable to believe that the jump of $\mathcal{O}(\lambda)$ in (40) will avoid in most cases performing NMLA along a circle where the number of branches are non-unique. However, this percussion may not always be enough since a new caustic curve may show up in a distance less than $2r$. The same conclusion applies if (38) and (39) fails in identifying a caustic curve.

If S_{q+1} is the first segment then one can create a first order polynom by using the NMLA-data from one point.

4.4 Algorithm 4, Sorting Algorithm.

Before one can call Algorithm 2 one must sort the data in each NMLA-point. I.e. the NMLA-algorithm only produce a set of βs and $\bar{u}_\lambda$s in each point and we do not know which pair $(\beta, \bar{u}_\lambda)_j$ in the sets

$$T_{min} : \left[(\beta, \bar{u}_\lambda)_1(\theta^{min}), \ldots (\beta, \bar{u}_\lambda)_{m(\theta^{min})}(\theta^{min})\right]$$

and

$$T_{max} : \left[(\beta, \bar{u}_\lambda)_1(\theta^{max}), \ldots (\beta, \bar{u}_\lambda)_{m(\theta^{max})}(\theta^{max})\right]$$

that belongs to the same branch. Note that $m(\theta^{min}) = m(\theta^{max})$ does not necessarily have to hold. We suggest the following sorting algorithm. Assume that we have m^{min} branches in θ^{min}, m^{max} branches in θ^{max} and that $m^{max} \leq m^{min}$, then the different possible *configurations* that we are bound to test for are set by the following:

- Store along rows in a matrix C all possible combinations to pick m^{max} elements from set T_{min}[5]. C is a
$$\frac{m^{min}!}{m^{max}!(m^{min}-m^{max})!} \times m^{max}$$
matrix.

- For each row ω_r^C in C create all permutations of the elements[6] and store them in rows in a matrix PC. PC is a
$$\frac{m^{min}!}{(m^{min}-m^{max})!} \times m^{max}$$
matrix.

- Create a configuration c_r by picking row ω_r^{PC} in PC, with length m^{max}, and associate element j in ω_r^{PC} to element j in set T_{max}. The m^{max} pairs of $(\beta, \bar{u}_\lambda)_j$ now define configuration c_r and hence there are
$$\frac{m^{min}!}{(m^{min}-m^{max})!}$$
configurations that are to be tested.

For each configuration c_r we compute $(\mathcal{P}_{(\phi,r,j)}, \mathcal{P}_{(A,r,j)})$ with Algorithm 2 for the matched pairs and with a first order approximation for the unmatched. We pick the configuration that minimize $|\tilde{u}_\lambda^{(r)}(\theta) - u(\theta)|_{L^1}$ approximately, i.e.

$$\min_r \sum_{\ell=1}^{\ell^{max}} \sum_{j=1}^{m_q} \left| \bar{u}_j^{(r)}(\theta_\ell) - u(\theta_\ell) \right|,$$

with $\theta_{q+1,\ell} = \theta_{q+1}^{min} + \ell\delta\theta \leq \theta_{q+1}^{max}$ for some $\delta\theta$.

If $m^{max} > m^{min}$ we proceed in a similar manner as above.

Below we use the following nomenclature. A wavefront which has folded over itself produces a set of new branches. Actually, analysis shows that from one wavefront you get, when it folds, a domain with three branches, next time it folds you get five branches and so on. However, when several wavefronts are present it is impossible to distinguish between that we have several wavefronts or several branches from one wavefront.

Assume that there are only one wave front present then we make the following remark. It would have been very attractive to use the knowledge that no caustic curve cross the segment since then we know that the relative "order of size" of the branches $\phi_1(\theta), \ldots \phi_m(\theta)$ are kept constant. I.e. if $\phi_3(\theta^{min}) < \phi_1(\theta^{min})$ holds then also $\phi_3(\theta^{max}) < \phi_1(\theta^{max})$ must hold. The sorting algorithm would have been trivial to implement in such a case. However, apart from that we may have several wavefronts we only know ϕ_j up to an integer ω_j (c.f. expression (24)) which will make this type of sorting un-feasible.

[5] In Matlab one can do *nchoosek*((1:m^{min}),m^{max}) to get such a matrix.
[6] In Matlab one can do *perm*((1:m^{max})) to get such a matrix.

4.5 Algorithm 5, Wavefront Propagation and Ray Tracing.

For inhomogeneous material the so called wavefront construction technique presented in [4] provide us with a method to propagate the solution along so called equi-phase fronts, $\phi(\tau)$ =constant, by propagating a set of markers along γ. However, the segments of phase functions $\phi_j^{(q)}(\theta)$, we have constructed with Algorithm 1-4 above, are clearly not in general equi-phase fronts. This is not of great importance and the methodology presented in [5] may be applied to these *generalized* wave fronts to.

For homogeneous material where the characteristic curves are straight, we have implemented the following algorithm.

Assume that we want to compute the GO-field in a point y. For each circle segment S_q, set by Algorithm 1-4, we can parameterize the angle

$$\alpha = \text{angle}(y - x(\theta)) \text{ where } x \in S_q,$$

by polynomial curve fitting and compute $\mathcal{P}_\alpha^{\kappa_\alpha}(\theta)$ for some κ_α. Hence we can search for the polynomial roots[7] to

$$\mathcal{P}_{(\alpha),q,j}^{\kappa_\alpha}(\theta) - \mathcal{P}_{(\beta),q,j}^{\kappa_\beta}(\theta) = 0. \tag{41}$$

For all $\zeta_{q,j}$ roots

$$\left[\theta_{1,q,j}, \ldots \theta_{\zeta_{q,j},q,j}\right],$$

fulfilling $\theta_q^{\min} \leq \theta < \theta_q^{\max}$, for all branches in all segments one can then compute the total GO-field $\tilde{u}_\lambda$ in y according to [3]

$$\sum_{q=1}^{\#S} \sum_{j=1}^{m_q} \sum_{k=1}^{\zeta_{q,j}} \mathcal{P}_{(A),q,j}(\theta_{k,q,j}) \sqrt{\frac{\mathcal{P}_{(\rho),q,j}(\theta_{k,q,j})}{\mathcal{P}_{(\rho),q,j}(\theta_{k,q,j}) + s(y,\theta_{k,q,j})}} \exp\left(\frac{i2\pi}{\lambda}(\mathcal{P}_{(\phi),q,j} + s(y,\theta_{k,q,j}))\right), \tag{42}$$

where $\#S$ is the number of segments given by Algorithm 3 and

$$s(y,\theta_{k,q,j}) = (y - R\left[\cos(\theta_{k,q,j}), \sin(\theta_{k,q,j})\right]) \cdot \left[\cos\left(\mathcal{P}_{(\beta),q,j}(\theta_{k,q,j})\right), \sin\left(\mathcal{P}_{(\beta),q,j}(\theta_{k,q,j})\right)\right].$$

We conclude that in the triple sum (42) we have implicitly used all polynoms defined in Section 1.1 ((11), (18), (20) and (22)), i.e. $\mathcal{P}_{(\phi)}$, $\mathcal{P}_{(\beta)}$, $\mathcal{P}_{(\rho)}$ and $\mathcal{P}_{(A)}$.

5 Optimization

We have seen two possibilities to optimize the coefficients in $\mathcal{P}_{(\phi)}$, $\mathcal{P}_{(\beta)}$, $\mathcal{P}_{(\rho)}$ and $\mathcal{P}_{(A)}$. What we call Global Optimization aims at reducing the size of the coefficients in front of the error terms in parametrical curve fitting, i.e. we will not alter the order of accuracy with optimization.

A second optimization, which we call Caustic Detection, aims at refining the position where the caustic curves $\chi_k(\tau)$ cross γ.

[7] In Matlab one can do *roots*($\left[c_3^\phi, c_2^\phi, c_1^\phi, c_0^\phi\right]$, $\left[c_{\kappa_\alpha}^\phi, \ldots, c_0^\phi\right]$) to get the roots.

5.1 Global Optimization

By using standard optimization algorithms provided in for example Matlab[8] one can minimize $|\tilde{u}_\lambda(\theta) - u(\theta)|_{L^1}$ by searching for local minimum of

$$\min_{\mathbf{c}} \sum_{\ell=1}^{\ell^{max}} \sum_{j=1}^{m_q} |\bar{u}_j(\theta_\ell, \mathbf{c}) - u(\theta_\ell)| \,,$$

with $\theta_{q,\ell} = \theta_q^{min} + \ell\delta\theta \leq \theta_q^{max}$ for some $\delta\theta$ and

$$\mathbf{c} = \left[c_{0,1}^{\phi} \ldots, c_{0,1}^{A} \ldots, c_{0,j}^{\phi}, \ldots c_{0,j}^{A} \ldots\right].$$

However, since $u(\theta)$ oscillates on the λ-scale we can not hope for that the order of accuracy in λ is improved since the error in the initial guess provided by Algorithm 1-4 are of $\mathcal{O}(\lambda)$. However, for each specific λ the accuracy may be improved. The increase of accuracy can also be achieved by increasing the order of the polynoms $\mathcal{P}_{(\phi)}$ and $\mathcal{P}_{(A)}$ where we note that the linear problems solved in such optimization may get ill-posed if there are too many polynom coefficients to be optimized.

5.2 Caustic Detection

As Algorithm 3 generally fails at identifying the caustic curves' exact position we have investigated the possibility to, by post-processing, refine their position on γ. This can be done as follows.

Let S_L and S_R be two neighboring segments with m_L branches and m_R branches respectively. Assume that Algorithm 3 found that the caustic curve crossed γ at $\theta = \theta_{\text{guess}}^{\text{caust.}}$. Then one can try to solve the following minimization problem

$$\min_{\theta^{\text{caust.}}} \left(\int_{\theta^{\text{caust.}}-\Delta\theta_L}^{\theta^{\text{caust.}}} |\tilde{u}_{\lambda,L}(\theta) - u(\theta)| \, d\theta + \int_{\theta^{\text{caust.}}}^{\theta^{\text{caust.}}+\Delta\theta_R} |\tilde{u}_{\lambda,R}(\theta) - u(\theta)| \, d\theta \right), \qquad (43)$$

where we can use extrapolation if we want to evaluate $\tilde{u}_{\lambda,L}$ and $\tilde{u}_{\lambda,R}$ outside their respective intervals on the circle γ. Note however, similar to Section 5.1, that u oscillates on the λ-scale and since the error in $\theta^{\text{caust.}}$ is of $\mathcal{O}(\lambda)$ the minimization algorithm may be trapped in a local minimum. Hence, to solve problem (43) we would need to use a global optimization algorithm and adaptive quadrature rules to be able to compute the integrals over the discontinuous norms.

6 Numerical Results

In this section we will illustrate the performance of our algorithms presented in Section 4. In those cases when we analyze the scattered field from circular Dirichlet boundary condition

[8] See for example in *lsqcurvefit()*.

we have used the analytic solution to compute $u(x)$ in CS. For a semi-circular geometry we have used a 2D Boundary Element Solver to compute the field on a Cartesian grid in CS. To access $u(x)$ with $x \in CS$ we used spline interpolation[9]. In all examples we have used $R = 2\text{m}$.

As already mentioned the algorithms needed to be tuned to improve the performance. This was done by modifying the size of the interval, indicated by the $\approx$-sign in (38) and by setting c_2. Also, the first initial guess of $\tilde{\theta}_{q+1}^{max}$ in the search algorithm was changed by modifying the constant c in

$$\tilde{\theta}_{q+1}^{max,(0)} = \theta_{q+1}^{min} + c \cdot (\theta_q^{max} - \theta_q^{min}),$$

as we changed Helmholtz solution.

6.1 No Boundary Conditions.

To simplify the numerical experiments we have initially tested our algorithms on a $u(x)$ that are not a solution to Helmholtz equation. In CS we set $u(x)$ to be

$$u(x(r,\theta)) = \sum_j^{m(\theta)} \exp\left(i\frac{2\pi}{\lambda} x \cdot [\cos(\beta_j), \sin(\beta_j)]\right),$$

where the angles $\beta_j(\theta), j = 1, \ldots, 3$ are given in the following table:

	β_1	β_2	β_3	$m(\theta)$
$-\frac{\pi}{4} \le \theta < \frac{\pi}{4}$	$\frac{7\pi}{4}$	0	$\frac{\pi}{4}$	3
$\frac{\pi}{4} \le \theta < \frac{\pi}{4} + \frac{\pi}{6}$	$\frac{\pi}{4}$	–	–	1
$\frac{\pi}{4} + \frac{\pi}{6} \le \theta < \frac{\pi}{4} + \frac{\pi}{3}$	$\frac{\pi}{4}$	$\frac{3\pi}{4}$	–	2
$\frac{\pi}{4} + \frac{\pi}{3} \le \theta < \frac{\pi}{4} + \frac{\pi}{2}$	$\frac{3\pi}{4}$	–	–	1
$\frac{\pi}{4} + \frac{\pi}{2} \le \theta < \frac{5\pi}{4}$	$\frac{3\pi}{4}$	π	$\frac{5\pi}{4}$	3
$\frac{5\pi}{4} \le \theta < \frac{5\pi}{4} + \frac{\pi}{6}$	$\frac{5\pi}{4}$	–	–	1
$\frac{5\pi}{4} + \frac{\pi}{6} \le \theta < \frac{5\pi}{4} + \frac{\pi}{3}$	$\frac{5\pi}{4}$	$\frac{7\pi}{4}$	–	2
$\frac{5\pi}{4} + \frac{\pi}{3} \le \theta < \frac{7\pi}{4}$	$\frac{7\pi}{4}$	–	–	1

(44)

6.1.1 Discontinuities and Sorting

In Figure 4 we illustrate in a "quiver"-plot the direction of propagation for the plane waves along γ. We plot arrows at θ_q^{min}, with $q = 1, \ldots, \#S$.

[9] C.f. *interp2()* in Matlab.

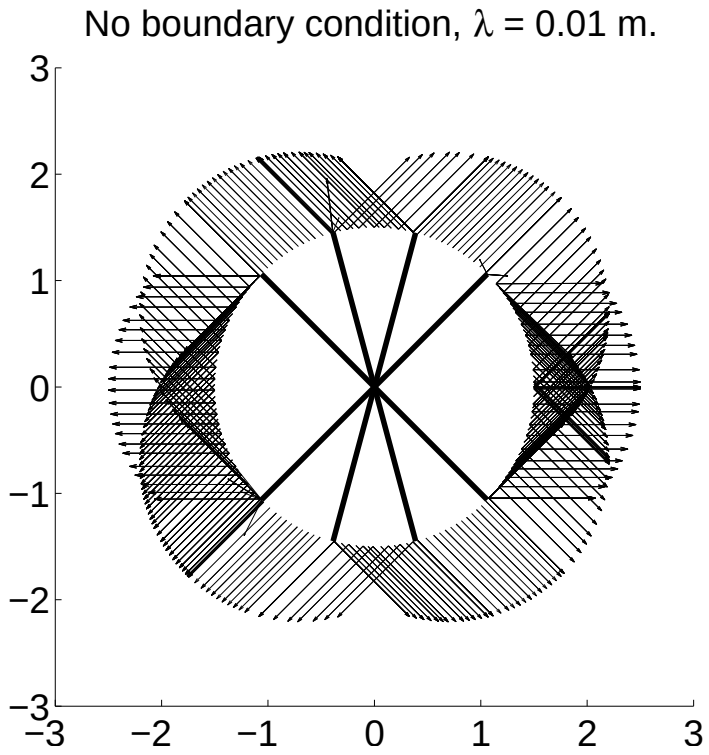

Figure 4: Illustration of rays associated to plane waves along the circle γ. The bold lines indicate the 8 sectors defined in table 44.

One can see, e.g. at $\theta = 7\pi/12$, that the discontinuity jumps sometimes coincide with the NMLA-analysis and produce defective βs.

In Figure 5 we illustrate the logarithmic error as a function of θ along γ. The *spikes* in the error plot originate from that the position of the caustic curves are only found within $\mathcal{O}(\lambda)$ distance.

In this example the $\approx$-sign in (38) has been implemented as the interval

$$\lambda \min_j |\bar{u}_j(\theta_{q+1}^{min})|\,[.9, 1] = [0.009, 0.01]\,.$$

Note, that this interval does only indirectly decide the error, since we use extrapolation with polynoms from previous segment in (38), while in Figure 5 we use interpolation of the polynoms associated to each specific segment.

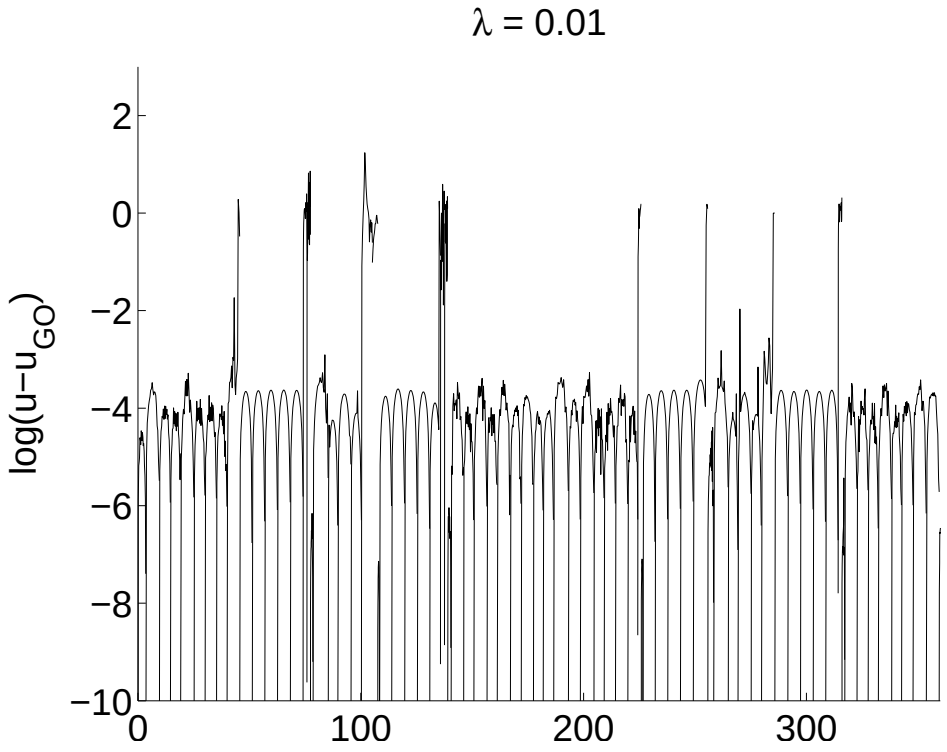

Figure 5: Illustration of error in the GO-solution on γ.

6.1.2 Convergence

We illustrate the $\mathcal{O}(\lambda)$-error in expression (31) by plotting the maximum error along γ between $\theta = 0$ and $\theta = 30\,\text{deg}$ in Figure 6. We clearly see that error is of $\mathcal{O}(\lambda)$.

In Figure 7 we try to illustrate that the width of the spike at $\theta = 35\,\text{deg}$ are of $\mathcal{O}(\lambda)$. By applying least squares to the numbers in the table below we achieve a spike width of $\mathcal{O}(\lambda^{0.98})$.

λ	$\Delta\theta$
10^{-2}	0.8
10^{-3}	0.1
10^{-4}	0.02
10^{-5}	0.0008

(45)

Hence, the numerical results confirms our expectations from the "jump" we do in (40).

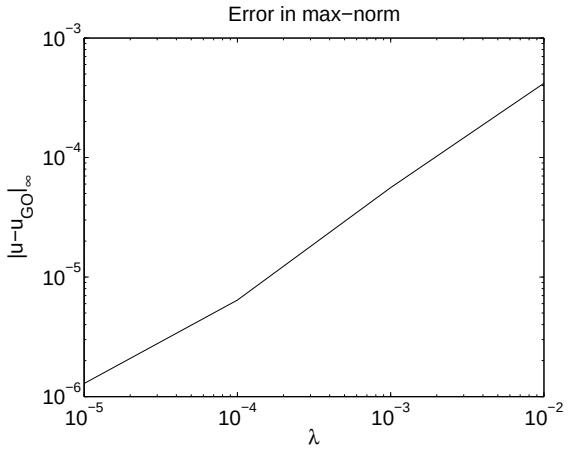

Figure 6: Convergence rate of the Wavefront Reconstruction algorithm.

6.1.3 Complexity

Since we use $\mathcal{P}^4_{(\phi)}$, $\mathcal{P}^3_{(A)}$ and a right hand side in (38) of $\mathcal{O}(\lambda)$ (c.f. discussion end of Section 2) we can expect that the complexity $\#S(\lambda)$ is of $\mathcal{O}(1/\sqrt{\lambda})$. This is confirmed by the following table which have been produced by computing the GO-field between $\theta = 0$ and $\theta = 30\,\mathrm{deg}$. We have

λ	$\#S$
10^{-2}	9
10^{-3}	24
10^{-4}	81
10^{-5}	228

(46)

which, by least squares, corresponds to $\#S(\lambda) = \mathcal{O}(\lambda^{-0.47})$. If a boundary element solver is used to compute u on γ then we need $\mathcal{O}(\lambda^{-1})$ floating point operations to set all polynoms in a segment. To set all segments on γ we hence need $\mathcal{O}(\lambda^{-1.5})$ floating point operations.

6.2 Dirichlet Boundary Condition on a Circle.

We excite the problem by letting a plane wave in the $\hat{x}_1$-direction, i.e.

$$u^{\mathrm{exc}}(x) = \exp\left(i\frac{2\pi}{\lambda}\left[\cos 0, \sin 0\right]\cdot x\right),$$

fall onto the scatterer. The boundary condition $u = 0$ is set on the unit circle, i.e. the scatterer.

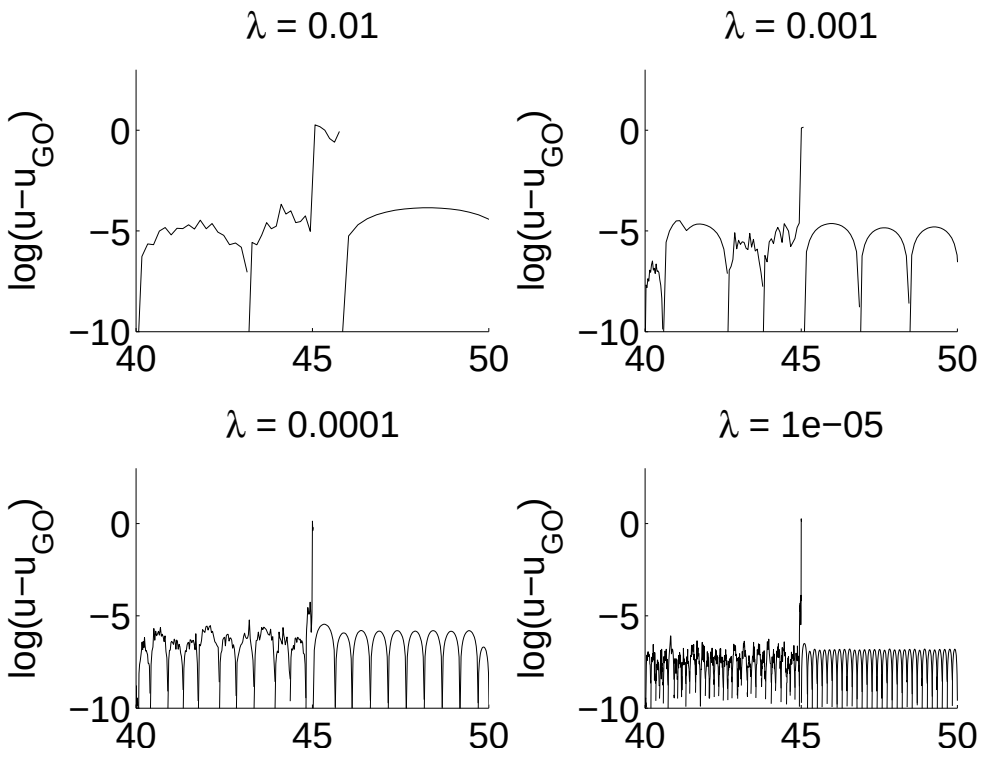

Figure 7: Illustration of the width of the spikes where our algorithm fails.

6.2.1 Convergence

We refer to Figure 8 and Figure 9. To illustrate what type of rays that may be captured by our algorithms we show the reflected and the so called Smooth Surface Diffracted [6] rays in a Dirichlet circle with radius 1. We capture the rays along two circle segments (reflection $50 < \theta < 310\,\mathrm{deg}$ and diffraction $(-40 < \theta < 40\,\mathrm{deg})$. The relative length of the arrows indicate the amplitude of the wavefront.

In Figure 10 we illustrate the anticipated errors for the radius of curvature ρ, the propagation direction β, the phase ϕ and u by plotting the error in a point $(\theta = \pi)$ on γ for the reflected rays. One clearly sees that pointwise error in ρ, β and u are of $\mathcal{O}(\lambda)$ while the error in ϕ is of $\mathcal{O}(\lambda^2)$. The numerical values are compared with analytic GO-solution to the scattering problem.

To illustrate that the $\mathcal{O}(\lambda)$ error in u is not destroyed when we solve (41) and apply (42) we plot in Figure 11 an estimation of the L_1 of u_{GO} along a circle segment with radius

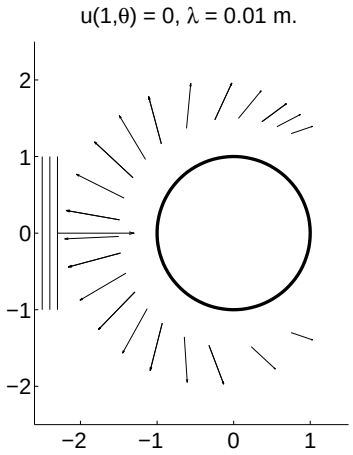

Figure 8: Illustration of reflected rays and u^{exc} are not included in the NMLA-analysis.

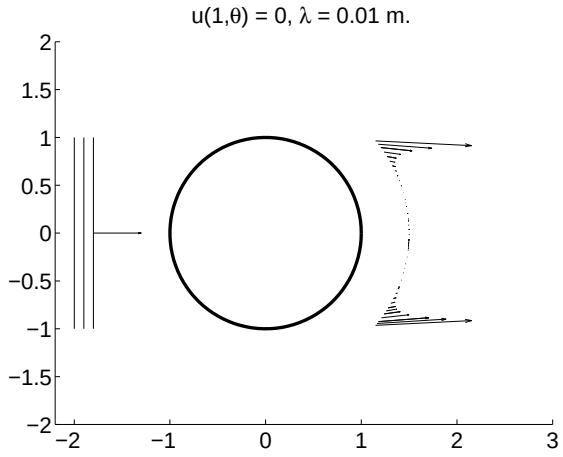

Figure 9: Illustration of Smooth Surface Diffracted rays. The arrows has been scaled with 5 for viewing purposes and u^{exc} are included in the NMLA-analysis.

2m. I.e. $\bar{u}$ has been propagated 0.5m by using (42) and is compared with the solution to Helmholtz equation.

6.2.2 Complexity

If a boundary element solver is used to compute u in CS then we need, according to Section 6.1.3, $\mathcal{O}(\lambda^{-1.5})$ floating point operations to set all polynoms in all segments. To compute

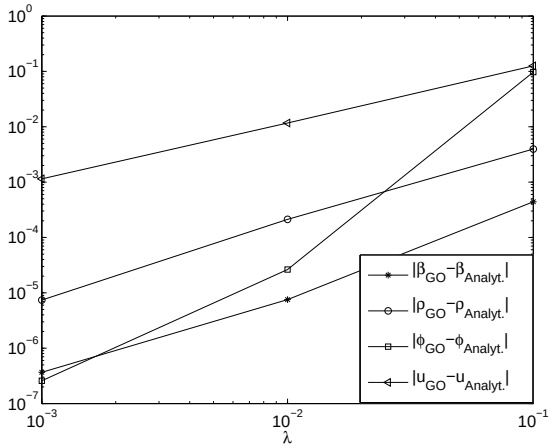

Figure 10: Illustration of errors in ρ, β, ϕ and u. The numerical values, indexed by GO, has been compared with analytical GO-solution in one point on γ.

u on a Cartesian grid with mesh size $\mathcal{O}(\lambda^{-1})$, in a square with side length of $\mathcal{O}(1)$ enclosing the scatterer, it would take $\mathcal{O}(\lambda^{-2}\lambda^{-0.5})$ floating point operations. The $\lambda^{-0.5}$-factor comes from the fact that we have to, for each grid point, solve $\mathcal{O}(\lambda^{-0.5})$ polynoms according to equation (41). Note however that this complexity could easily be decreased by increasing the number of degrees for $\mathcal{P}_{(\phi)}$, $\mathcal{P}_{(A)}$ and $\mathcal{P}_{(\beta)}$.

The $\mathcal{O}(\lambda^{-2.5})$ complexity could be compared with if we compute u in each grid point by evaluating a boundary integral over the scatterer. The complexity is in this case of $\mathcal{O}(\lambda^{-3})$.

6.3 Dirichlet Boundary Condition on a Semi-Circle.

We excite the problem by letting the plane wave

$$u^{\text{exc}}(x) = \exp\left(i\frac{2\pi}{\lambda}\hat{x}_1 \cdot x\right),$$

fall onto the scatterer. The boundary condition $u = 0$ is set on the semi-circle, i.e. $u(1, -\pi < \theta < \pi) = 0$.

To illustrate that edge diffracted rays ($\bar{u} = \mathcal{O}(\sqrt{\lambda})$) in the semi-circle are captured by the algorithm, we refer to Figure 12. To illustrate the failure of the NMLA-algorithm, c.f. discussion in Section 4.1, when there is not enough of separation in β-space, we refer to Figure 13. Also, since there the number of branches probably changes quite often along γ it may be that the jump in (40) are not successful and hence NMLA is called too close to a caustic curve. We capture the rays along two circle segments (edge diffraction $45 < \theta < 90\,\text{deg}$ and reflection $(160 < \theta < 200\,\text{deg})$.

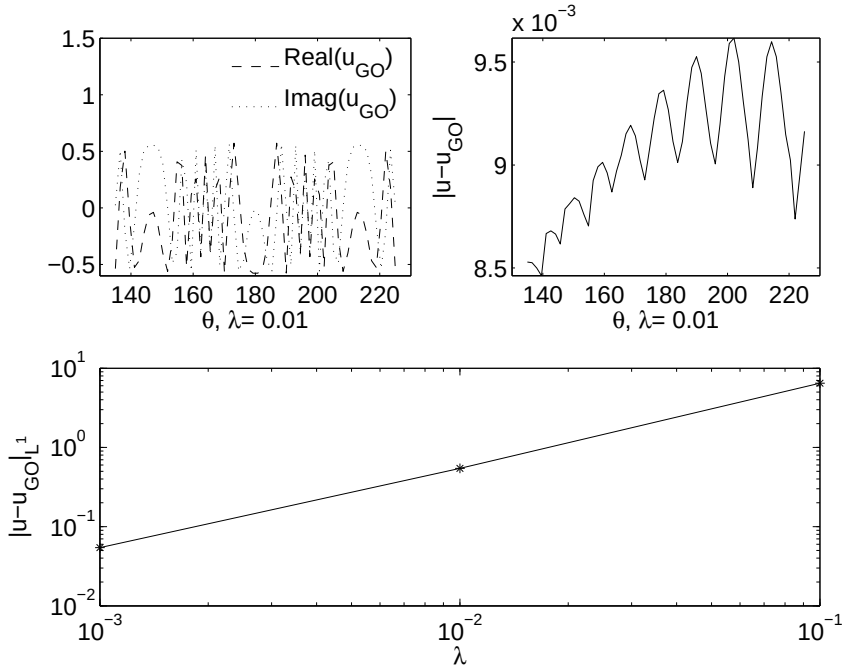

Figure 11: The L_1 error of the propagated reflected $\bar{u}$ along a circle segment at radius $r = 2$m. Convergence is of $\mathcal{O}(\lambda)$. u^{exc} are not included in the NMLA-analysis.

7 Future Work and Discussion

We suggest that, in future work, the failure of the wavefront reconstruction algorithm close to caustics are solved by interpolation. I.e. we know, by applying the error measure (31), the domains where the algorithm fails then one could blend the solution to the left and to right of the caustic curve in this domain.

In our algorithm there exist some parameters that should be adjusted to achieve optimal performance. This is only practical if there are some a priori knowledge about the scattered field. To obtain a more general wavefront reconstruction algorithm the parameters should adaptively be tuned as the algorithm steps along γ.

We also suggest that more effort is put into the NMLA-algorithm itself to improve it so that it can distinguish between two branches a and b with the same β but with different phase, i.e.

$$\log\left(\frac{\lambda\bar{u}_a}{i2\pi|\bar{u}_a|}\right) \neq \log\left(\frac{\lambda\bar{u}_b}{i2\pi|\bar{u}_b|}\right).$$

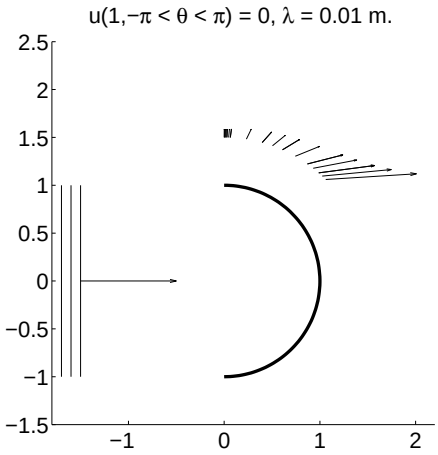

Figure 12: Illustration of edge diffracted rays in a semi-circle. u^{exc} are not included in the NMLA-analysis.

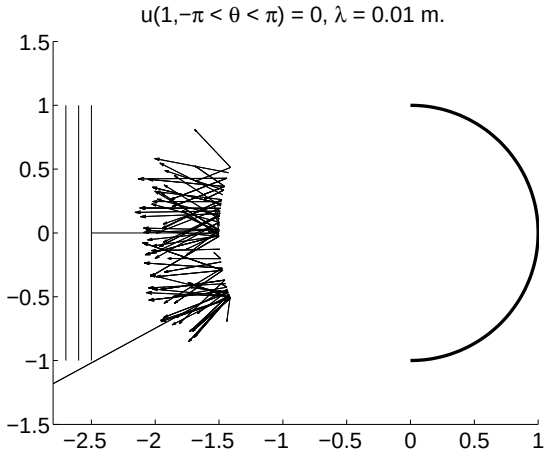

Figure 13: Illustration of a problem that seems to be too complex for our algorithms. We see reflected and diffracted rays in a semi-circle and that the NMLA-algorithm fails since many of the rays along γ are not sufficiently separated in β-space. Also the jump in (40) is probably not successful in avoiding using the NMLA on caustic curves. u^{exc} are not included in the NMLA-analysis.

The optimal choice of R has up until now not been discussed. It will depend on what type solver and what type of boundary conditions that are analyzed.

Of course if a FEM of FD solver is used to compute u in CS then, from a an efficiency point of view, one wants to pick R as small as possible. If BEM-solver is used then the number of floating point operations to compute u in CS is not dependent on R. However, assuming that $\mathbb{R}^2 \setminus \Sigma$ are homogeneous, the following criterias must be balanced from the wavefront construction algorithm point of view:

- If NMLA-points are close to the scatterer, i.e. R small, then intuitively it is more probable that rays are well separated in β-space and the NMLA-algorithm will therefore be able to distinguish them.
- If NMLA-points are far from the scatterer, i.e. R large, then intuitively it is more probable that assumption 3 and expression (5) are fulfilled.

As indicated by the numerical experiments in section 6.1 the complexity of finding the asymptotic solution on a closed curve γ depends on how well we want the discontinuities to be resolved. If we a priori know that there are no discontinuities then the number of segments can be decreased by increasing the degrees of the polynoms $\mathcal{P}$. This is naturally a gain compared to rigorous computation of u along γ where we always need to resolve the wavelength. We also want to emphasize that it is computationally cheap, relative to for example a boundary integral evaluation, to propagate a GO solution from γ.

As indicated by the results in Section 6.1.2 the width of the "error spikes" are of $\mathcal{O}(\lambda)$. Hence, this width decreases faster than the segment size which by the results in Section 6.1.3 are of size $\mathcal{O}(\sqrt{\lambda})$. It then follows that the L_1 error is of $\mathcal{O}(\lambda)$.

We have not yet investigated the possibilities to use the wavefront reconstruction algorithms for specific applications. E.g we believe that one may exploit the knowledge of the phase if one would like to do shape optimization, with respect to the scattered field, of the scatterer Σ or if one wants to interpolate between solutions for different incoming waves.

Another application to the wavefront reconstruction algorithm would be to apply it to Smooth Surface Diffraction (SSD). I.e. there also exist an asymptotic theory for the rays, or the surface field, that propagate along smooth surfaces. If one can, from knowledge of local surface geometrical quantities and the incoming field, design an NMLA-algorithm then one could improve existent [5] wavefront construction algorithms for SSD.

Acknowledgments The work were funded by Institut National de Recherche en Informatique et en Automatique (INRIA) and Parallel and Scientific Computing Institute (PSCI/KTH).

The author acknowledge Fabrice DELBARY for adapting his two dimensional Boundary Element Solver and for letting us using it.

The author also would like to thank Jean-David BENAMOU, Francis COLLINO and Olof RUNBORG for many valuable ideas.

References

[1] J. -D. Benamou, F. Collino and O. Runborg. Numerical Microlocal Analysis of Harmonic Wavefields. *J. Comput. Phys.*, 199:717–741, 2004.

[2] J. B. Keller. Geometrical Theory of Diffraction. *J. opt. soc. of Amer.*, 52(2):61–72, February 1962.

[3] J. B. Keller and R. M. Lewis. Asymptotic Methods for Partial Differential Equations: The Reduced Wave Equation and Maxwell's Equations. Course given at UCLA during winter quarter 1999.

[4] V. Vinje, E. Iversen and H. Gjöystdal. Traveltime and Amplitude Estimation Using Wavefront Construction. *Geophysics*, 58:1157–1166, 1993.

[5] S. Hagdahl. *Hybrid Methods for Computational Electromagnetics in the Frequency Domain.* PhD thesis No. TRITA-NA-0441, Stockholm University, February 2005.

[6] B. R. Levy and J. B. Keller. Diffraction by Smooth Objects. *Communications on pure and applied mathematics*, pages 159–209, 1959.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399